(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開2003－168284
（P2003－168284A）
(43)公開日　平成15年６月13日(2003.6.13)

(51)Int.Cl.'	識別記号	ＦＩ		テーマコード（参考）
Ｇ１１Ｂ　27/034		Ｇ１１Ｂ　20/10	Ｇ	５Ｃ０５３
20/10		20/12		５Ｄ０４４
20/12		27/00	Ｄ	５Ｄ１１０
27/00		27/02	Ｋ	
Ｈ０４Ｎ　5/91		Ｈ０４Ｎ　5/91	Ｎ	

審査請求　未請求　請求項の数３　ＯＬ　（全 22 頁）

(21)出願番号　特願2001－363589(P2001－363589)

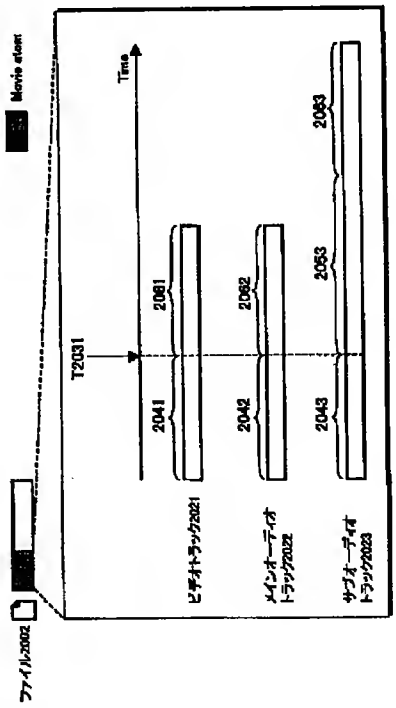
(22)出願日　平成13年11月29日(2001.11.29)

(71)出願人　000005049
シャープ株式会社
大阪府大阪市阿倍野区長池町22番22号
(72)発明者　木山　次郎
大阪府大阪市阿倍野区長池町22番22号　シャープ株式会社内
(72)発明者　岩野　裕利
大阪府大阪市阿倍野区長池町22番22号　シャープ株式会社内
(74)代理人　100102277
弁理士　佐々木　晴康　（外２名）

最終頁に続く

(54)【発明の名称】　データ記録方法およびデータ編集方法

(57)【要約】
【課題】　サブオーディオの付加されたビデオを編集する際のユーザの手間を削減する方法を提供することを目的とする。
【解決手段】　第1のデータと第2のデータとを関係付けて記録する記録方法であって、前記第2のデータを、前記第1のデータの特定部分との同期が必要な第1の種別と、前記第1のデータの特定部分との同期が不要な第2の種別とに区別して記録する。すなわち、ビデオトラックに対するサブオーディオトラックの関係（独立／依存）を示す情報を用意し、その情報に基づきビデオトラック編集時のサブオーディオトラックの扱いを変える。



1

【特許請求の範囲】
【請求項１】　第1のデータと第2のデータとを関係付けて記録する記録方法であって、
前記第2のデータを、前記第1のデータの特定部分との同期が必要な第1の種別と、前記第1のデータの特定部分との同期が不要な第2の種別とに区別して記録することを特徴とするデータ記録方法。
【請求項２】　前記請求項１に記載の記録方法に従って記録されたデータの編集方法であって、
前記第1のデータの編集に際して、前記第1の種別に区別された第2のデータは、関係する第1のデータと同時に編集することを特徴とするデータ編集方法。
【請求項３】　前記請求項１に記載の記録方法に従って記録されたデータの編集方法であって、
前記第1のデータの編集に際して、前記第1の種別に区別された第2のデータは、関係する第1のデータと同時に編集しないことを特徴とするデータ編集方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、ハードディスク、光ディスク等のランダムアクセス可能な記録媒体に対して、映像データ、音声データを記録・編集するデータ記録方法及びデータ編集方法に関するものである。
【０００２】
【従来の技術】ディスクメディアを用いたビデオのディジタル記録再生装置（以下、ビデオディスクレコーダと呼ぶ）が普及しつつある。テープメディアにはないディスクメディアにおける特徴機能として、非破壊編集機能あるいはノンリニア編集機能と呼ばれるものがある。この機能は、ディスク上に記録したAVストリームを移動あるいはコピーすることなく、AVストリームの任意の区間（シーン）を任意の順番で再生できる、というもので、AVストリームのどこからどこまでどういう順番で再生するかを示す情報(再生管理情報)を作り、その情報に従って再生することで実現される。
【０００３】例えば、図３０に示すように、あるコンテンツがビデオデータと、前記ビデオデータと同時に収録し、前記ビデオデータと同期して再生されるべきオーディオデータと、前記ビデオデータおよびオーディオデータとは別に後から追加したオーディオデータとの3種類のデータで構成され、それぞれをトラックとして、Video track3000、Main audio track3010、Sub audio track3020の3トラックで管理していたとする。
【０００４】このとき、図中の区間T3051からT3052の区間を削除しようとした場合、ユーザの望むであろう結果として次の2種類が考えられる。第1の結果は、T3051からT3052の区間に対応する3トラックの区間すべて（図中の区間3002、3012、3022）が再生対象からはずれ、図３１のように、その前後の区間が連続的に再生されることである。

2

【０００５】このような結果は、再生対象からはずされる区間の関連が強い場合、例えばSub audio track3020の区間3022がVideo track3000の区間3002に対する英語での吹き替えであった場合に望まれると考えられる。なぜなら、Sub audio track3020の区間3022が再生対象から外されなかった場合、対応するビデオの無い英語の吹き替え音声のみが残るからである。
【０００６】一方、ユーザの望むであろう第2の結果は、T3051からT3052の区間に対応するVideo track3000およびMain audio track3001の区間（図中の区間3002、3012）のみ再生対象から外れ、図３２のように、Video track3000およびMain audio track3010に関しては、その前後の区間が連続的に再生され、Sub audio track3020に関しては削除前と変わらず再生されることである。
【０００７】このような結果は、Sub audio track3020のデータの連続性が高く、他のトラックとの関連が弱い場合、例えばSub audio track3020がVideo track3000に対するBGMであった場合に望まれると考えられる。なぜなら、Sub audio track3020の区間3022が再生対象からはずされた場合、BGMが不連続になるためである。
【０００８】従来のテープメディアを用いた編集の場合、第1の結果を得るのは、コピーの際、T3051からT3052を除くだけでよいため比較的容易であるが、第2の結果を得ようとすると、まず、T3051からT3052を除いてコピーし、再度Sub audio trackのデータを録音する必要があり、二度手間となる。それに対してディスクメディアでは、トラックの管理情報を書き換えるだけで第1、第2の結果を容易に得ることができる。
【０００９】
【発明が解決しようとする課題】しかしながら、従来の技術において、前記第1の結果を得ようとすると、ユーザは図３０に示すような画面から区間3002、3012、3022をそれぞれ選択し、削除コマンドを実行することになる。同様に、前記第2の結果を得ようとすると、ユーザは図３０に示すような画面から区間3002、3012を選択し、削除コマンドを実行することになる。すなわち、どこを削除するかを個別に明示的に指定する必要がある。
【００１０】本発明は、上記課題を鑑みてなされたものであり、サブオーディオの付加されたビデオを編集する際のユーザの手間を削減することが可能なデータ記録方法及びデータ編集方法を提供することを目的とする。
【００１１】
【課題を解決するための手段】本願の第１の発明は、第1のデータと第2のデータとを関係付けて記録する記録方法であって、前記第2のデータを、前記第1のデータの特定部分との同期が必要な第1の種別と、前記第1のデータの特定部分との同期が不要な第2の種別とに区別して記録することを特徴とする。
【００１２】本願の第２の発明は、前記第1のデータの編集に際して、前記第1の種別に区別された第2のデータ

が、関係する第1のデータと同時に編集されることを特徴とする。

【００１３】本願の第３の発明は、前記第1のデータの編集に際して、前記第1の種別に区別された第2のデータが、関係する第1のデータと同時に編集されないことを特徴とする。

【００１４】

【発明の実施の形態】以下、本発明の実施形態について、図面を参照しながら詳細に説明する。ここでの説明は、本発明において共通に用いる構成、個々の実施形態に固有の内容という順に行っていく。

【００１５】<システム構成>図１は本発明において共通に用いる、アフレコ可能なビデオディスクレコーダの構成図である。この装置は、図１に示すように、バス100、ホストCPU101、RAM102、ROM103、ユーザインタフェース104、システムクロック105、光ディスク106、ピックアップ107、ECC(Error Correctinq Coding)デコーダ108、ECCエンコーダ109、再生用バッファ110、記録/アフレコ用バッファ111、デマルチプレクサ112、マルチプレクサ113、多重化用バッファ114、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118、および図示しないカメラ、マイク、スピーカ、ディスプレイ等で構成される。

【００１６】ホストCPU101は、バス100を通じてデマルチプレクサ112、マルチプレクサ113、ピックアップ107、また図示していないが、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118との通信を行う。

【００１７】再生時に、光ディスク106からピックアップ107を通じて読み出されたデータは、ECCデコーダ108によって誤り訂正され、再生用バッファ110に一旦蓄えられる。デマルチプレクサ112はオーディオデコーダ115、ビデオデコーダ116からのデータ送信要求に従い、再生用バッファ中のデータをその種別によって適当なデコーダに振り分ける。

【００１８】一方、記録時に、オーディオエンコーダ117とビデオエンコーダ118によって圧縮符号化されたデータは、多重化用バッファ114に一旦送られ、マルチプレクサ113によってAV多重化され、記録/アフレコ用バッファ111に送られる。記録/アフレコ用バッファ111中のデータは、ECCエンコーダ109によって誤り訂正符号を付加され、ピックアップ107を通じて光ディスク106に記録される。

【００１９】オーディオデータの符号化方式にはMPEG-1 Layer-IIを、ビデオデータの符号化方式にはMPEG-2をそれぞれ用いる。

【００２０】光ディスク106は、外周から内周に向かって螺旋状に記録再生が行われる脱着可能な光ディスクとする。2048byteを1セクタとし、誤り訂正のため16セクタでECCブロックを構成する。ECCブロック中のデータを

書き換える場合、そのデータが含まれるECCブロック全体を読み込み、誤り訂正を行って、対象のデータを書き換え、再び誤り訂正符号を付加し、ECCブロックを構成して、記録媒体に記録する必要がある。また、光ディスク106は、記録効率を上げるためZCAV（ゾーン角速度一定）を採用しており、記録領域は回転数の異なる複数のゾーンで構成される。

【００２１】<ファイルシステム>光ディスク106上の各種情報を管理するためにファイルシステムを用いる。ファイルシステムには、パーソナルコンピュータ（PC）との相互運用を考慮してUDF（Universal Disk Format）を使用する。ファイルシステム上では、各種管理情報やAVストリームはファイルとして扱われる。

【００２２】ユーザエリアは、2048byteの論理ブロック（セクタと一対一対応）で管理される。各ファイルは、整数個のエクステント（連続した論理ブロック）で構成され、エクステント単位で分散して記録しても良い。空き領域は、Space Bitmapを用いて論理ブロック単位で管理される。

【００２３】<ファイルフォーマット>AVストリーム管理のためのフォーマットとして、QuickTimeファイルフォーマットを用いる。QuickTimeファイルフォーマットとは、Apple社が開発したマルチメディアデータ管理用フォーマットであり、PCの世界で広く用いられている。

【００２４】QuickTimeファイルフォーマットは、ビデオデータやオーディオデータ等（これらを総称してメディアデータとも呼ぶ）と管理情報とで構成される。両者を合わせてここでは、QuickTimeムービー（略してムービー）と呼ぶ。両者は同じファイル中に存在しても、別々のファイルに存在しても良い。

【００２５】同じファイル中に存在する場合は、図２（a）に示すような構成をとる。各種情報はatomという共通の構造に格納される。管理情報はMovie atomという構造に格納され、AVストリームはMovie data atomという構造に格納される。尚、Movieatom中の管理情報には、メディアデータ中の任意の時間に対応するAVデータのファイル中での相対位置を導くためのテーブルや、メディアデータの属性情報や、後述する外部参照情報等が含まれている。

【００２６】一方、管理情報とメディアデータを別々のファイルに格納した場合は、図２（b）に示すような構成をとる。管理情報はMovie atomという構造に格納されるが、AVストリームはatomには格納される必要はない。このとき、Movie atomはAVストリームを格納したファイルを「外部参照」している、という。

【００２７】外部参照は、図２（c）に示すように、複数のAVストリームファイルに対して行うことが可能であり、この仕組みにより、AVストリーム自体を物理的に移動することなく、見かけ上編集を行ったように見せる、いわゆる「ノンリニア編集」「非破壊編集」が可能にな

る。

【００２８】それでは、図３乃至図１２を用いて、QuickTimeの管理情報のフォーマットについて説明する。まず、共通の情報格納フォーマットであるatomについて説明する。atomの先頭には、そのatomのサイズであるAtom size、そのatomの種別情報であるTypeが必ず存在する。Typeは4文字で区別され、例えばMovie atomでは'moov'、Movie data atomでは'mdat'となっている。

【００２９】各atomは別のatomを含むことができる。すなわち、atom間には階層構造がある。Movie atomの構成を図３に示す。Movie header atomは、そのMovie atomが管理するムービーの全体的な属性を管理する。Track atomは、そのムービーに含まれるビデオやオーディオ等のトラックに関する情報を格納する。User data atomは、独自に定義可能なatomである。

【００３０】Track atomの構成を図４に示す。Track header atomは、そのトラックの全体的な属性を管理する。Edit atomは、メディアデータのどの区間を、ムービーのどのタイミングで再生するかを管理する。Track reference atomは、別のトラックとの関係を管理する。Media atomは、実際のビデオやオーディオといったデータを管理する。User data atomは、メーカー独自定義の情報を管理する。本発明においては、このatomを含むトラックの種別、例えばオリジナルトラック、サブオーディオトラック等に関する情報を格納する。

【００３１】Track header atomの構成を図５に示す。ここでは、後での説明に必要なもののみについて説明する。flagsは属性を示すフラグの集合である。代表的なものとして、Track enabledフラグがあり、このフラグが1であれば、そのトラックは再生され、0であれば再生されない。layerはそのトラックの空間的な優先度を表しており、画像を表示するトラックが複数あれば、layerの値が小さいトラックほど画像が前面に表示される。また、Track IDは、このatomを内包するトラックのIDを示し、同じIDのトラックは1個のムービー内には複数存在しない。

【００３２】Media atomの構成を図６に示す。Media header atomは、そのMedia atomの管理するメディアデータに関する全体的な属性等を管理する。Handler referenceatomは、メディアデータをどのデコーダでデコードするかを示す情報を格納する。Media information atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。

【００３３】Media information atomの構成を図７に示す。Media information header atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。Handler reference atomは、Media atomの項で説明した通りである。Data information atomは、そのQuickTimeムービーが参照するメディアデータを含むファイルの名前を管理するatomであるData reference atomを含む。Sample table atomは、データのサイズや再生時間等を管理している。

【００３４】次に、Sample table atomについて説明するが、その前に、QuickTimeにおけるデータの管理方法について、図８を用いて説明する。QuickTimeでは、データの最小単位（例えばビデオフレーム）をサンプルと呼ぶ。個々のトラック毎に、サンプルには再生時間順に1から番号（サンプル番号）がついている。

【００３５】また、QuickTimeフォーマットでは、個々のサンプルの再生時間長およびデータサイズを管理している。また、同一トラックに属するサンプルが再生時間順にファイル中で連続的に配置された領域をチャンクと呼ぶ。チャンクにも再生時間順に、1から番号がついている。

【００３６】さらに、QuickTimeフォーマットでは、個々のチャンクのファイル先頭からのアドレスおよび個々のチャンクが含むサンプル数を管理している。これらの情報に基づき、任意の時間に対応するサンプルの位置を求めることが可能となっている。

【００３７】Sample table atomの構成を図９に示す。Sample description atomは、個々のチャンクのデータフォーマット（Data format）やサンプルが格納されているファイルのチャンクの Index等を管理する。Time-to-sample atomは、個々のサンプルの再生時間を管理する。

【００３８】Sync sample atomは、個々のサンプルのうち、デコード開始可能なサンプルを管理する。Sample-to-chunk atomは、個々のチャンクに含まれるサンプル数を管理する。Sample size atomは、個々のサンプルのサイズを管理する。Chunk offset atomは、個々のチャンクのファイル先頭からのアドレスを管理する。

【００３９】Edit atomは、図１０に示すように、1個のEdit list atomを含む。Edit listatomはNumber of entriesで指定される個数分の、Track duration、Media time、Media rateの値の組（エントリ）を持つ。各エントリは、トラック上で連続的に再生される区間に対応し、そのトラック上での再生時間順に並んでいる。

【００４０】Track durationはそのエントリが管理する区間のトラック上での再生時間、Media timeはそのエントリが管理する区間の先頭に対応するメディアデータ上での位置、Media rateはそのエントリが管理する区間の再生スピードを表す。尚、Media timeが-1の場合は、そのエントリのTrack duration分、そのトラックでのサンプルの再生を停止する。この区間のことをempty editと呼ぶ。

【００４１】図１１にEdit listの使用例を示す。ここでは、Edit list atomの内容が図１１（a）に示す内容であり、さらにサンプルの構成が図１１（b）であったとする。尚、ここではi番目のエントリのTrack durationをD(i)、Media timeをT(i)、Media rateをR(i)とす

る。このとき、実際のサンプルの再生は、図１１（c）に示す順に行われる。このことについて簡単に説明する。

【００４２】まず、エントリ#1はTrack durationが13000、Media timeが20000、Media rateが1であるため、そのトラックの先頭から13000の区間はサンプル中の時刻20000から33000の区間を再生する。次に、エントリ#2はTrack durationが5000、Mediatimeが−1であるため、トラック中の時刻13000から18000の区間、何も再生を行わない。

【００４３】最後に、エントリ#3はTrack durationが10000、Media timeが0、Media rateが1であるため、トラック中の時刻18000から28000の区間において、サンプル中の時刻0から10000の区間を再生する。

【００４４】図１２にTrack reference atomの構成を示す。このatomは、このatomを内包するトラックと他のトラックとの関係を管理している。トラック間の関係には、複数のタイプがあり、タイプ毎にTrack reference type atomを設ける。Track reference type atomの構成について説明する。typeは、このatomが管理する関係のタイプ、例えばタイプトラック間の同期を示す“sync”やチャプター区切りを示す“chap”等を格納する。track-IDsにはこのatomを内包するトラックと、typeで示される関係にあるトラックのTrack IDを格納する。

【００４５】図１３にUser data atomの構成を示す。このatomには、QuickTimeフォーマットで定義されてない独自の情報を任意個数格納することができる。1個の独自情報は1個のエントリで管理され、1個のエントリはSizeとTypeとUser dataで構成される。Sizeはそのエントリ自体のサイズを表し、Typeは独自情報をそれぞれ区別するための識別情報、User dataは実際のデータを表す。

【００４６】＜AVストリームの形態＞まず、本実施例におけるAVストリームの構成について、図１４及び図１５を用いて説明する。AVストリームは整数個のRecord Unit（RU）で構成される。RUはディスク上で連続的に記録する単位である。RUの長さは、AVストリームを構成するRUをどのようにディスク上に配置してもシームレス再生（再生中に画像や音声が途切れないで再生できること）やリアルタイムアフレコ（アフレコ対象のビデオをシームレス再生しながらオーディオを記録すること）が保証されるように設定される。この設定方法については後述する。

【００４７】また、RU境界がECCブロック境界と一致するようにストリームを構成する。RUのこれらの性質によって、AVストリームをディスクに記録した後も、シームレス再生を保証したまま、ディスク上でRU単位の配置を容易に変更することができる。

【００４８】RUは、整数個のVideo Unit（VU）で構成される。VUは単独再生可能な単位であり、そのことから再生の際のエントリ・ポイントとなりうる。

【００４９】VU構成を図１５に示す。VUは、1秒程度のビデオデータを格納した整数個のGOP（グループ・オブ・ピクチャ）と、それらと同じ時間に再生されるメインオーディオデータを格納した整数個のAAU（オーディオ・アクセス・ユニット）とから構成される。

【００５０】尚、GOPは、MPEG-2ビデオ規格における画像圧縮の単位であり、複数のビデオフレーム（典型的には15フレーム程度）で構成される。AAUはMPEG-1 Layer-II規格における音声圧縮の単位で、1152点の音波形サンプル点により構成される。サンプリング周波数が48kHzの場合、AAUあたりの再生時間は0.024秒となる。VU中では、AV同期再生のために必要となる遅延を小さくするため、AAU、GOPの順に配置する。

【００５１】また、VU単位で独立再生を可能とするために、VU中のビデオデータの先頭にはSequence Header（SH）を置く。VUの再生時間は、VUに含まれるビデオフレーム数にビデオフレーム周期をかけたものと定義する。さらに、VUを整数個組み合わせてRUを構成する場合、RUの始終端をECCブロック境界に合わせるため、VUの末尾を0で埋める。

【００５２】＜AVストリーム管理方法＞AVストリームの管理方法は、前述のQuickTimeファイルフォーマットをベースにしている。図１６にAVストリーム管理形態を示す。ビデオトラックは、各ビデオフレームを1サンプル（ビデオサンプル）、VU中のビデオの塊を1チャンク（ビデオチャンク）として管理する。メインオーディオトラックは、AAUを1サンプル（オーディオサンプル）、VU中のオーディオの塊を1チャンク（オーディオチャンク）として管理する。

【００５３】＜RU単位決定方法＞次に、RU単位決定方法について説明する。この決定方法では、基準となるデバイス(リファレンス・デバイス・モデル)を想定し、その上でシームレス再生が破綻しないように連続記録単位を決める。

【００５４】それではまず、リファレンス・デバイス・モデルについて、図１７を用いて説明する。リファレンス・デバイス・モデルは１個のピックアップとそれにつながるECCエンコーダ・デコーダ501、トラックバッファ502、デマルチプレクサ503、アフレコ用バッファ504、オーディオエンコーダ509、ビデオバッファ505、オーディオバッファ506、ビデオデコーダ507、オーディオデコーダ508とによって構成される。

【００５５】本モデルにおけるシームレス再生は、VUのデコード開始時にトラックバッファ502上に少なくとも1個VUが存在すれば保証されるものとする。オーディオフレームデータのECCエンコーダ501へのデータの入力速度およびECCデコーダ501からデータの出力速度はRsとする。

【００５６】また、アクセスによる読み出し、記録の停

止する最大期間をTaとする。さらに、短いアクセス(100トラック程度)に要する時間をTkとする。なお、これら期間には、シーク時間、回転待ち時間、アクセス後に最初にディスクから読み出したデータがECCから出力されるまでの時間が含まれる。本実施例では、Rs=20Mbps、Ta=1秒、Tk=0.2秒とする。

【００５７】前記リファレンス・デバイス・モデルにおいて再生を行った場合、次のような条件を満たせば、トラックバッファ502のアンダーフローがないことが保証できる。

【００５８】条件を示す前にまず、記号の定義を行う。AVストリームを構成するi番目の連続領域をC#iとし、C#i中に含まれる再生時間をTc(i)とする。Tc(i)はC#i中に先頭が含まれているVUの再生時間の合計とする。また、C#iからC#i+1へのアクセス時間をTaとする。

【００５９】また、再生時間Tc(i)分のVU読み出し時間をTr(i)とする。このとき、トラックバッファ502をアンダーフローさせない条件とは、分断ジャンプを含めた最大読み出し時間をTr(i)としたとき、任意のC#iにおいて、

Tc(i)≧Tr(i)+Ta・・・＜式１＞

が成立することである。

【００６０】なぜなら、この式は、シームレス再生の十分条件である、

【００６１】

【数１】

$$\sum_{i} Tc(i) \geq \sum_{i} (Tr(i) + Ta)$$

【００６２】を満たす十分条件であるためである。

【００６３】＜式1＞中のTr(i)に、Tr(i)=Tc(i)×(Rv+Ra)/Rsを代入して、Tc(i)で解くとシームレス再生を保証可能なTc(i)の条件

Tc(i)≧(Ta×Rs)/(Rs-Rv-Ra)・・・＜式２＞

が得られる。

【００６４】つまり、各連続領域に先頭の含まれるVUの合計が上式を満たすようにすれば、シームレス再生を保証可能である。このとき、各連続領域には合計の再生時間が上式を満たす完全なVU群を含むように制限しても良い。

【００６５】自動分割ムービーファイルでも＜式２＞を満たす必要がある。ただし、先頭の自動分割ムービーの最初のRUおよび末尾の自動分割ムービーの最後のRUは＜式２＞を満たさなくてもよい。なぜなら、先頭は記録媒体からのデータ読み出し開始より再生開始を遅らせることにより吸収でき、末尾については次に続くデータがないため連続再生を気にする必要が無いからである。このように先頭と末尾において条件を緩めることにより、短い空き領域を有効利用できる。

【００６６】＜インデックス・ファイル＞ディスク内に含まれるQuickTimeムービーや静止画データ等を含む各種コンテンツ（以後、AVファイルと呼ぶ）を管理するため、AV Indexファイルという特別のQuickTimeムービーファイルをディスク内に１個置く。図１８に、AV Indexファイルの構成を示す。AV Indexファイルは通常のQuickTimeムービーファイルと同様、管理情報であるMovie atom1791とデータ自体のMovie data atom1792で構成される。

【００６７】AV Indexファイルは、複数のエントリを管理し、ディスク内の各AVファイルはそれぞれ1個のエントリで管理される。さらに、各AVファイルをまとめるための入れ物（以後フォルダと呼ぶ）等もそれぞれ1個のエントリで管理する。

【００６８】Movie atom1791は、各エントリの属性情報を管理するためのProperty track1793、各エントリのタイトル文字列データを管理するためのTitle track1794、各エントリのサムネイル画像データを管理するためのThumbnail track1795、各エントリの代表オーディオデータを管理するためのIntro music track1796の計4種類のトラックで構成される。

【００６９】各エントリに関する属性情報、はそれぞれの1792～1795のトラックのサンプルとして管理される。例えばAVファイル1740に関する属性情報はProperty track1793上のサンプル1701、タイトル文字列データはTitle track1794上のサンプル1711、サムネイル画像データはThumbnail track1795上のサンプル1721、代表オーディオデータはIntro music track1796上のサンプル1731で管理する。

【００７０】サンプル間の対応付けは、各サンプルの再生開始時間に基づき行う。すなわち、トラック間で同一時刻に位置するサンプルが同一エントリに対応していると判断する。

【００７１】Movie data atom1793は、各AVファイルに関する属性情報や、タイトル文字列データ、サムネイル画像データ、代表オーディオデータを格納する。属性情報は図１９に示す構成を取る。各フィールドについて説明する。versionは、ファイルフォーマットのバージョンを示す。pe-flagsは各種フラグをまとめたものであり、詳細は後述する。

【００７２】parent-entry-numberは、属性情報に対応するエントリが属するフォルダに対応するエントリのentry-numberを格納、entry-numberは、属性情報に対応するエントリのentry-numberを格納する。この2個の情報で、ファイルとフォルダの包含関係を表す。set-dependent-flagsおよびuser-private-flagsについては、説明を省略する。

【００７３】creation-timeおよびmodification-timeはこの属性情報に対応するエントリが作成された日時、修正された日時を表す。durationはこの属性情報に対応するエントリの再生時間を表す。binary-file-identifierは、この管理情報に対応するエントリがファイルに対応

していた場合、そのファイルのパス名を固定長にエンコーディングしたもので、詳細についての説明は省略する。

【００７４】referred-counterはこの属性情報に対応するエントリが管理するファイルが他のファイルから参照されている回数を格納する。referring file listは、実際に参照しているファイルのパス名のリストを格納する。URL file identifierは、管理するファイルが上記のbinary-file-identifierにエンコードできない場合にURL（Unified Resource Locator）形式で、ファイルのパスを格納する。

【００７５】＜記録時の処理＞ユーザから録画が指示された場合の処理を、図２０に沿って説明する。このとき記録するAVストリームは、ビデオのビットレートRv=5Mbps、オーディオのサンプリング周波数48kHz、ビットレートRa=Rp=256kbpsであるものとする。すでに、ファイルシステムの管理情報はRAM102上に読み込まれているとする。

【００７６】まず、ストリームの構成や連続領域の構成を決定する(ステップ701)。1VUを1GOP=30フレームで構成するとしたとき、＜式４＞にRs=20Mbps、Ta=1秒、Rv=5Mbps、Ra=256kbpsを代入し、Te(i)の範囲である1.36秒以上が得られる。1VUの再生時間を0.5秒としているため、RU再生時間は2秒とする。

【００７７】次に、ムービーファイル記録準備を行う。具体的にはファイルをopenし、1個のRUを連続的に記録可能な空き領域を探す。存在しなければ録画を中止し、録画できないことをユーザに知らせる(ステップ702)。

【００７８】また、オーディオエンコーダ117、ビデオエンコーダ118をそれぞれ起動する(ステップ703)。次に、記録用バッファ111に１RU分のデータが蓄積されているかどうかチェックする(ステップ704)。

【００７９】蓄積されていれば、記録用バッファ111中の１RU分のデータを連続的に記録する（ステップ705)。蓄積されていなければ、記録終了が指示されていないかどうかチェックし（ステップ706）、指示されていなければステップ704を実行し、指示されていれば以下の記録終了処理を行う。

【００８０】まず、現在記録中のムービーファイルに残りデータ（ステップ707）および管理情報を記録する（ステップ708）。最後に、AV Indexファイルに今回作成したQuickTimeムービーファイルを登録する（ステップ709）。なお、ビデオトラックとオーディオトラックのトラック種別は「オリジナル」に設定する。

【００８１】＜サブオーディオ付加時の処理＞既に録画したビデオに対して、オーディオ（サブオーディオ）データを付加する際の処理について、図２１乃至図２４を用いて説明する。ユーザからのサブオーディオデータ付加方法として次の2種類が考えられる。

（１）新規に入力したオーディオデータを付加（アフレコ）

（２）既記録のオーディオデータを指定して付加

それぞれについて以下で説明を行う。

【００８２】＜サブオーディオ付加時の処理：新規入力オーディオを付加＞まず、新規に入力したオーディオデータを付加する、いわゆるアフレコを行う場合について説明する。図２１に示すように、光ディスク106上には、ビデオデータとオーディオデータの格納されたQuickTimeムービーファイル2002と前記QuickTimeムービーファイルの登録されたAV Indexファイル2001が記録されているものとする。

【００８３】QuickTimeムービーファイル2002に格納されているビデオデータおよびオーディオデータは、それぞれビデオトラック2011とメインオーディオトラック2012で管理されているものとする。このときQuickTimeムービーファイル2002に対して、新規に入力するオーディオデータをサブオーディオとして付加する場合の処理を、図２２に示すフローチャートを用いて説明する。

【００８４】まず、初期状態として再生可能なQuickTimeムービーファイルのリストをGUI画面上に表示する（ステップ801）。このリストはAV Indexファイル2001に記録されている情報を元に作成されたものである。ユーザはムービーファイル2002にサブオーディオを付加したいため、前記リストからQuickTimeムービーファイル2002を選択し、再生を指示し、その結果、ムービーファイル2002の再生を開始する（ステップ802）。

【００８５】次に、ユーザはアフレコを開始したい個所で一時停止を指示し、再生を一時停止する（ステップ803）。次に、ユーザはアフレコ開始を指示し、その結果アフレコを開始する（ステップ804）。そのとき、アフレコオーディオデータを記録するためのファイルとして、図２３に示すようにQuickTimeムービーファイル2003を新規作成する。

【００８６】アフレコ中はオーディオエンコーダ117から入力されるオーディオデータをQuickTimeムービーファイル2003に記録する（ステップ805）。ユーザからアフレコ停止が指示されたら、入力されたオーディオデータに関する管理情報をMovie atomの形でファイル2003に格納する（ステップ806）。次に、追加するサブオーディオを管理するトラック2023をQuickTimeムービーファイル2002に作成する（ステップ807）。

【００８７】トラック2023のトラック種別は「サブオーディオ」に設定する。その直後に、ユーザに対して「ビデオに従属（ビデオと同期して編集される）」「ビデオと独立（ビデオの編集と無関係）」の選択画面を提示する（ステップ808）。新規入力の場合、ビデオに従属したものである可能性が高いため、初期値は「ビデオに従属」に設定しておく。

【００８８】ユーザが「ビデオに従属」を選択した場合、まず、トラック2023のTrack header atom中にTrack

reference atomにtypeが“sync”のTrack reference type atomを追加し、そのTrack reference type atomのTrack IDsにQuick Timeムービーファイル2002のビデオトラック2021のTrack IDを格納する（ステップ810）。

【００８９】次に、ユーザに対して、入力されたオーディオデータを、別のコンテンツでも再利用する可能性があるかどうかを確認し（ステップ811）、可能性がある場合、ムービーファイル2003をAV Indexファイル2001に登録する（ステップ812）。

【００９０】次に、そのトラック2023に対し、ムービーファイル2003の情報を元にSample table atomを追加する。このとき、データの参照先はムービーファイル2003とする。さらに、そのトラックに対し、サブオーディオの再生開始点、つまり現在一時停止している個所を元にEdit list atomを作成する（ステップ813）。

【００９１】最後に、RAM102上にあるQuickTimeムービーファイル2002の管理情報を光ディスク106上のQuickTimeムービーファイル2002のMovie atomに反映させ（ステップ814）、光ディスク106上のAV Indexファイル2001中のQuickTimeムービーファイル2002に対応するエントリのReferred-countを1増加させる（ステップ815）。

【００９２】<サブオーディオ付加時の処理：既記録オーディオを付加>次に、既記録のオーディオデータを指定して付加する場合について説明する。図２４に示すように、光ディスク106上には、ビデオデータとオーディオデータの格納されたQuickTimeムービーファイル2102と、オーディオデータのみを格納したQuickTimeムービーファイル2103と、前記QuickTimeムービーファイルの登録されたAV Indexファイル2101が記録されているものとする。

【００９３】QuickTimeムービーファイル2102に格納されているビデオデータおよびオーディオデータは、それぞれビデオトラック2121とメインオーディオトラック2112で管理されているものとする。また、QuickTimeムービーファイル2103に格納されているオーディオデータは、それぞれオーディオトラック2122で管理されているものとする。

【００９４】このときQuickTimeムービーファイル2102にQuickTimeムービー2103をサブオーディオとして付加する場合の処理を、図２５に示すフローチャートを用いて説明する。

【００９５】まず、初期状態として再生可能なQuickTimeムービーファイルのリストをGUI画面上に表示する（ステップ901）。このリストはAV Indexファイル2101に記録されている情報を基に作成されたものである。ユーザはムービーファイル2102にサブオーディオを付加したいため、前記リストからQuickTimeムービーファイル2102を選択し、再生を指示する。その結果、ムービーファイル2102の再生を開始する（ステップ902）。

【００９６】次に、ユーザはサブオーディオを付加したい個所で一時停止を指示し、再生を一時停止する（ステップ903）。次に、サブオーディオ付加を指示し、その指示に応じて、AV Indexファイル2101に登録されているオーディオデータのみのQuickTimeムービーファイルのリストをGUI画面に表示する（ステップ904）。このリストの中からユーザはムービーファイル2103を選択する。

【００９７】次に、追加するサブオーディオを管理するトラック2123をQuickTimeムービーファイル2102に作成する（ステップ905）。トラック2123のトラック種別は「サブオーディオ」に設定する。その直後に、ユーザに対して「ビデオに従属（ビデオと同期して編集される）」「ビデオと独立（ビデオの編集と無関係）」の選択画面を提示する（ステップ906）。なお、ファイルから選択した場合は前記のアフレコと比べ同期させない可能性が高いため、初期値は「ビデオと独立」に設定しておく。

【００９８】ユーザが「ビデオに従属」を選択した場合、トラック2123のTrack header atom中にTrack reference atomにtypeが“sync”のTrack reference type atomを追加し、そのTrack reference type atomのTrack IDsにQuick Timeムービーファイル2102のビデオトラック2121のTrack IDを格納する（ステップ908）。

【００９９】次に、トラック2123に対し、QuickTimeムービーファイル2103の情報を元にSample table atomを追加する。このとき、データの参照先はムービーファイル2103とする。さらに、そのトラックに対し、サブオーディオの再生開始点、つまり現在一時停止している個所を元にEdit list atomを作成する（ステップ909）。

【０１００】最後に、RAM102上にあるQuickTimeムービーファイル2002の管理情報を光ディスク106上のQuickTimeムービーファイル2002のMovie atomに反映させ（ステップ910）、光ディスク106上のAV Indexファイル2101中のQuickTimeムービーファイル2102に対応するエントリのReferred-countを1増加させる（ステップ911）。

【０１０１】<削除処理>本実施形態における、部分削除時の処理について、図２６を例にとって説明する。ここでは、ユーザから図２６に示すようにQuickTimeムービーファイル2002の部分区間T2031～T2032の削除が指示されたとする。このときの処理を、図２７に示すフローチャートを用いて説明する。

【０１０２】まず、削除対象のQuickTimeムービーファイルにサブオーディオトラックが存在するかどうかチェックする（ステップ1001）。存在しなければステップ1004を実行する。存在したなら、そのサブオーディオトラック(トラック2023)がビデオトラック（トラック2021）と同期しているかどうかを調べる（ステップ1002）。

【０１０３】具体的にはTrack header atom中のTrack reference atomにtypeが“sync”のTrack reference type atomが存在し、track-IDsにビデオトラックのTrack IDが含まれていれば、同期していると判断する。同期し

ていなければステップ1004を実行し、同期していれば、部分区間T2031～T2032に対応するサブオーディオトラックの区間（区間2053）が再生対象から除外される（ステップ1003）。

【０１０４】次に、ビデオトラックおよびメインオーディオトラック中の部分区間T2031～T2032に対応する区間（区間2051および2052）が再生対象から除外される（ステップ1004）。次に、上記の編集結果を光ディスク106上のAVファイル（QuickTimeムービーファイル2002）のMovie atomに反映させる（ステップ1005）。

【０１０５】最後に、上記の編集結果を光ディスク106上のAV Indexファイル（AV Indexファイル2001）に反映させる。具体的には、編集に伴うDurationやModification-time等の変化を反映させる。

【０１０６】以上の処理の結果、サブオーディオトラック2023がビデオトラック2021と同期していた場合、図２８に示すように、ビデオトラック2021、メインオーディオトラック2022、サブオーディオトラック2023について、それぞれ区間2051、2052、2053が再生対象から除外される。

【０１０７】一方、サブオーディオトラック2023がビデオトラック2021と同期していなかった場合、ビデオトラック2021、メインオーディオトラック2022のみ、それぞれ図２９に示すように、区間2051、2052が再生対象から除外され、サブオーディオトラック2203は削除前と変わりがない。図２８の場合、ビデオトラック2021の区間2061以降の区間は、サブオーディオトラック2023のみ再生されることになり、不自然になるため、ビデオトラック2021の区間2061以降の区間に対応するサブオーディオトラック2023の区間を再生対象から外してもよい。

【０１０８】つまり、ユーザはサブオーディオを記録する際に一度属性を指定するだけで、編集時に特に指定することなく、自分の意図した編集（ビデオを部分削除することによって、BGMが不連続になったり、吹き替え音声だけが残ったりすることがない）が可能となっている。

【０１０９】ここでは説明は省略するが、T2011にビデオを挿入する場合も上記の説明と同様、サブオーディオトラック2023がビデオトラック2021と同期している場合、サブオーディオトラック2023のT2031の区間に無再生区間を挿入し、同期していない場合、サブオーディオトラック2023を変更しないのは言うまでもない。

【０１１０】また、QuickTimeムービーファイル2002の削除がユーザから指示された場合、サブオーディオトラック2023から参照されているQuickTimeムービーファイル2003がAV Indexファイル2001に登録されていなければ、QuickTimeムービーファイル2003は再利用される可能性がないため、QuickTimeムービーファイル2002の削除と共にQuickTimeムービーファイル2003も削除する。QuickTimeムービーファイル2003がAV Indexファイル2001に登録されていれば、QuickTimeムービーファイル2002のみ削除する。

【０１１１】さらに、本実施形態では、部分削除を、QuickTimeムービーファイル2002のMovie atom、すなわち管理情報の書き換えのみで実現したが、QuickTimeムービーファイル2002、2003に含まれる実際のAVデータを削除することが可能なことは言うまでもない。

【０１１２】＜バリエーション＞本実施形態では、サブオーディオトラックとビデオトラックとの関係をTrackreference atomの情報で管理しているが、両トラックの関係を示せるものであれば、どのような形態であってもよいことは言うまでもない。例えば、AV Indexファイルのような、別ファイルで管理されていても構わない。

【０１１３】また、本実施形態においては、サブオーディオ付加時に、ユーザに「ビデオと独立」「ビデオに依存」を選択させているが、システムが自動設定しても構わない。

【０１１４】さらに、本実施形態では、サブオーディオを取り上げているが、本発明はサブオーディオにのみ限定されない。例えば、ビデオに重畳表示するアニメーションに対しても適用可能であることは言うまでもない。

【０１１５】そしてまた、本実施形態においては、サブオーディオデータを参照元のムービーファイルとは別のファイルに格納しているが、同一のファイルに格納しても構わない。

【０１１６】また、本実施形態においては、外部参照するオーディオデータファイルを、QuickTimeムービーファイルフォーマットで記録しているが、それ以外のフォーマット、たとえばMP3(MPEG 1 Audio Layer 3)フォーマットやWAVフォーマット（Windows（登録商標）における標準オーディオフォーマット）であってもよい。ただし、サブオーディオトラックにおいてそれらのオーディオデータファイルを参照する際には、Sample table atomを構築するために、それらのファイルを解析する必要がある。

【０１１７】

【発明の効果】以上説明したように、本発明によれば、ビデオデータに対してサブオーディオデータを後から付加した際に、ビデオとの同期関係を指定することにより、削除、追加等の編集時にユーザは特に指定することなく、自分の意図した編集が可能となる。

【図面の簡単な説明】

【図１】本発明の実施形態における概略構成を示すブロック図である。

【図２】QuickTimeファイルフォーマットにおける管理情報とAVストリームとの関係を示す説明図である。

【図３】QuickTimeファイルフォーマットにおけるMovie atomの概要を示す説明図である。

【図４】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図５】QuickTimeファイルフォーマットにおけるTrack header atomの構成を示す説明図である。
【図６】QuickTimeファイルフォーマットにおけるMedia atomの構成を示す説明図である。
【図７】QuickTimeファイルフォーマットにおけるMedia information atomの構成を示す説明図である。
【図８】Sample table atomによるデータ管理の例を示す説明図である。
【図９】QuickTimeファイルフォーマットにおけるSample table atomの構成を示す説明図である。
【図１０】QuickTimeファイルフォーマットにおけるEdit atomの構成を示す説明図である。
【図１１】Edit atomによる再生範囲指定の例を示す説明図である。
【図１２】QuickTimeファイルフォーマットにおけるTrack reference の構成を示す説明図である。
【図１３】QuickTimeファイルフォーマットにおけるUser data atomの構成を示す説明図である。
【図１４】本発明におけるAVストリームの構成を示す説明図である。
【図１５】本発明におけるVUの構造を示す説明図である。
【図１６】本発明におけるAVストリーム管理形態を示す説明図である。
【図１７】本発明におけるリファレンス・デバイス・モデルを示す説明図である。
【図１８】本発明におけるAV Indexの構成を示す説明図である。
【図１９】本発明におけるAV Index中の属性情報の構成を示す説明図である。
【図２０】本発明における、録画動作を示すフローチャートである。
【図２１】本発明における、サブオーディオデータ付加直前の状態の例を示す説明図である。
【図２２】本発明における、新規入力サブオーディオ付加動作を示すフローチャートである。
【図２３】本発明における、サブオーディオデータ付加直後の状態の例を示す説明図である。
【図２４】本発明における、既記録データをサブオーディオとして付加する直前の状態の例を示す説明図である。
【図２５】本発明における、既記録サブオーディオデータ付加動作を示すフローチャートである。
【図２６】本発明における、部分削除直前の状態の例を示す説明図である。
【図２７】本発明における、部分削除動作を示すフローチャートである。
【図２８】本発明における、部分削除後における第１の状態の例を示す説明図である。
【図２９】本発明における、部分削除後における第２の状態の例を示す説明図である。
【図３０】従来技術における、部分削除直前の状態の例を示す説明図である。
【図３１】従来技術における、ユーザの望むであろう部分削除後の第１の結果の例を示す説明図である。
【図３２】従来技術における、ユーザの望むであろう部分削除後の第２の結果の例を示す説明図である。

【符号の説明】

100 バス
101 ホストCPU
102 RAM
103 ROM
104 ユーザインタフェース
105 システムクロック
106 光ディスク
107 ピックアップ
108 ECCデコーダ
109 ECCエンコーダ
110 再生用バッファ
111 記録/アフレコ用バッファ
112 デマルチプレクサ
113 マルチプレクサ
114 多重化用バッファ
115 オーディオデコーダ
116 ビデオデコーダ
117 オーディオエンコーダ
118 ビデオエンコーダ

【図３】

```
Movie atom {
        Atom size
        Type(='moov')
        Movie header atom
        Track atom (video track)
        Track atom (main audio track)
              :
        User data atom
}
```

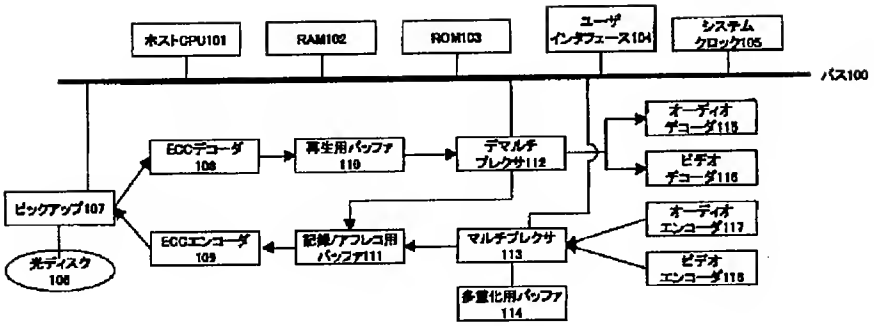
【図４】

```
Track atom {
        Atom size
        Type('trak')
        Track header  atom
        Edit atom
        Track reference atom
        Media atom
        User data atom
              :
}
```

【図６】

```
Media atom {
        Atom size
        Type('mdia')
        Media header  atom
        Handler reference atom
        Media information atom
        User data atom
              :
}
```

【図１】



【図５】

```
Track header atom {
        Atom size
        Type(='tkhd')
        Version
        Flags
        Creation time
        Modification time
        Track ID
        Reserved
        Duration
        Reserved
        Layer
        Alternate group
        Volume
        Reserved
        Matrix structure
        Track width
        Track height
}
```

【図７】

```
Media information atom {
        Atom size
        Type(='minf')
        {Video or Sound or Base} media information header atom
        Handler reference atom
        Data information atom
        Sample table atom
}
```

【図８】

```
Sample table atom {
        Atom size
        Type(='stbl')
        Sample description atom
        Time-to-sample atom
        Sync sample atom
        Sample-to-chunk atom
        Sample size atom
        Chunk offset atom
}
```

【図１０】

```
Edit atom {
        Atom size
        Type('edts')
        Edit list atom
}

Edit list atom {
        Atom size
        Type('elst')
        Versions
        Flags
        Number of entries(=N)
        for (i = 0; i < N; i++){
                Track duration
                Media time
                Media rate
        }
}
```

【図１２】

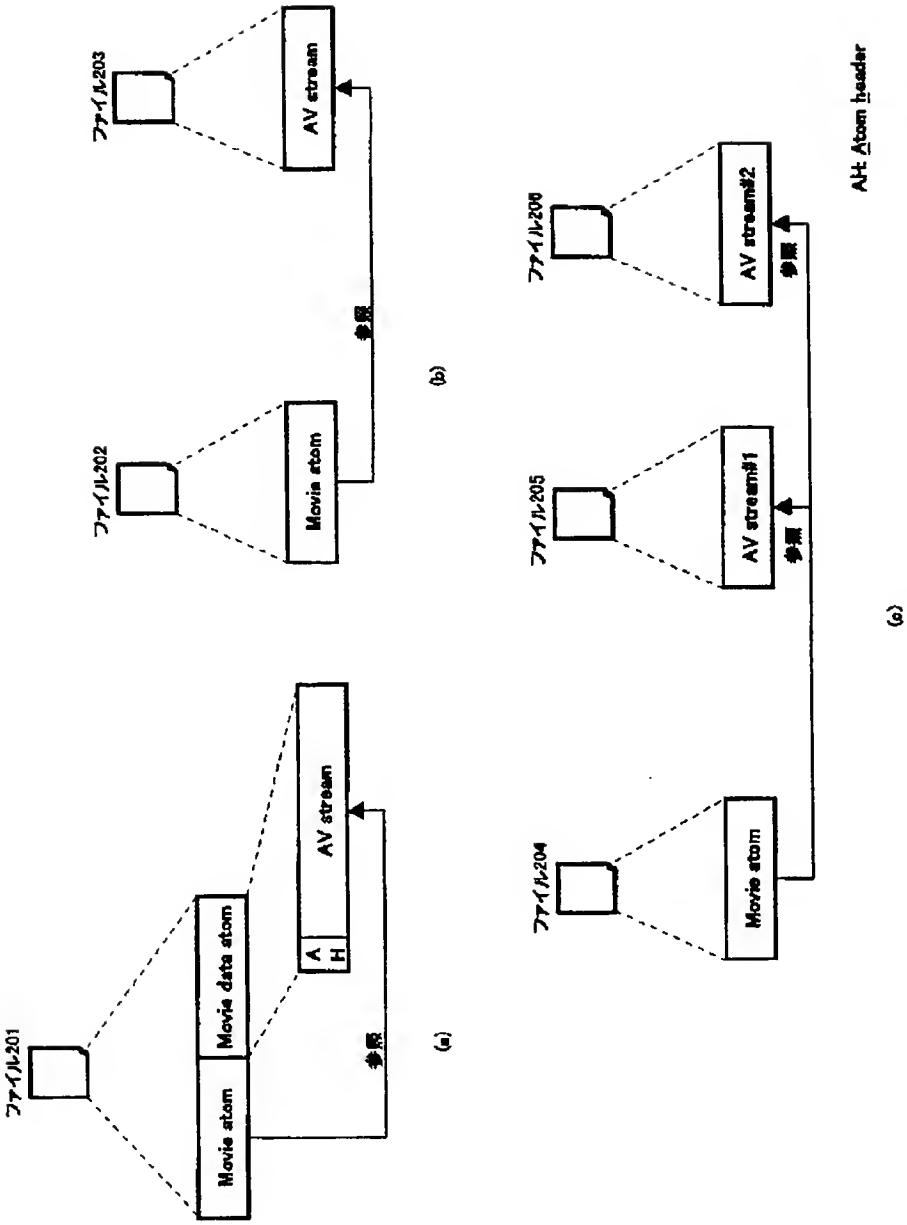
```
Track reference atom {
        Atom size
        Type('tref')
        for (i = 0; i < N; i++){
                Track reference type atom
        }
}

Track reference type atom {
        Atom size
        Type
        for (j = 0; j < M; j++){
                track-IDs
        }
}
```

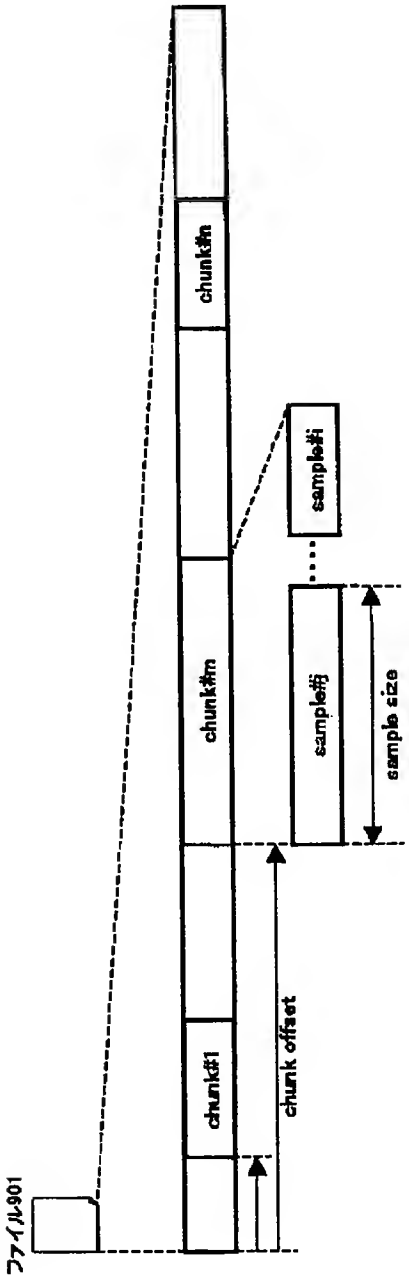
【図１３】

```
User data atom {
        Atom size
        Type('udta')
        for (i=0;i<N; i++){
                Atom size
                Type
                User data
        }
}
```

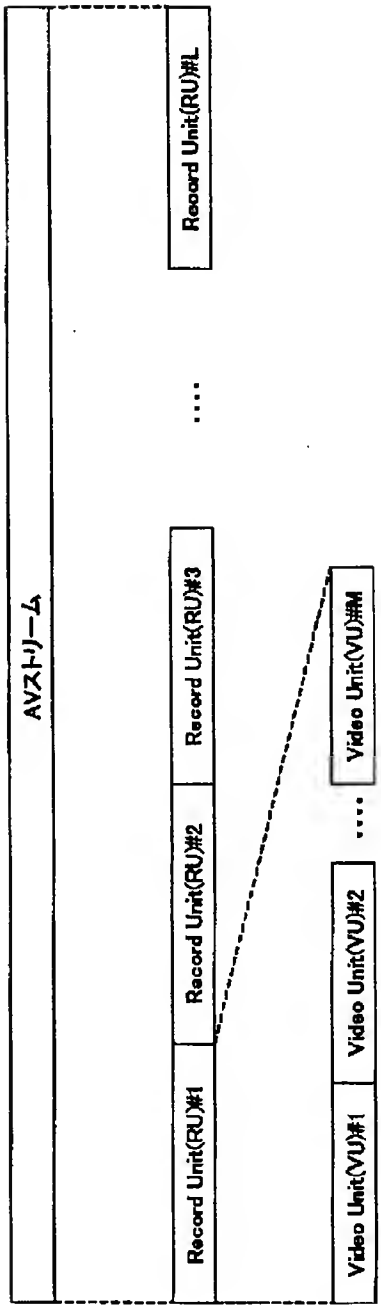
【図２】


ファイル201
Movie atom
Movie data atom
AV stream
A H
参照
(a)
ファイル202
Movie atom
ファイル203
AV stream
参照
(b)
ファイル204
Movie atom
ファイル205
AV stream#1
ファイル206
AV stream#2
参照
AH: Atom header
(c)

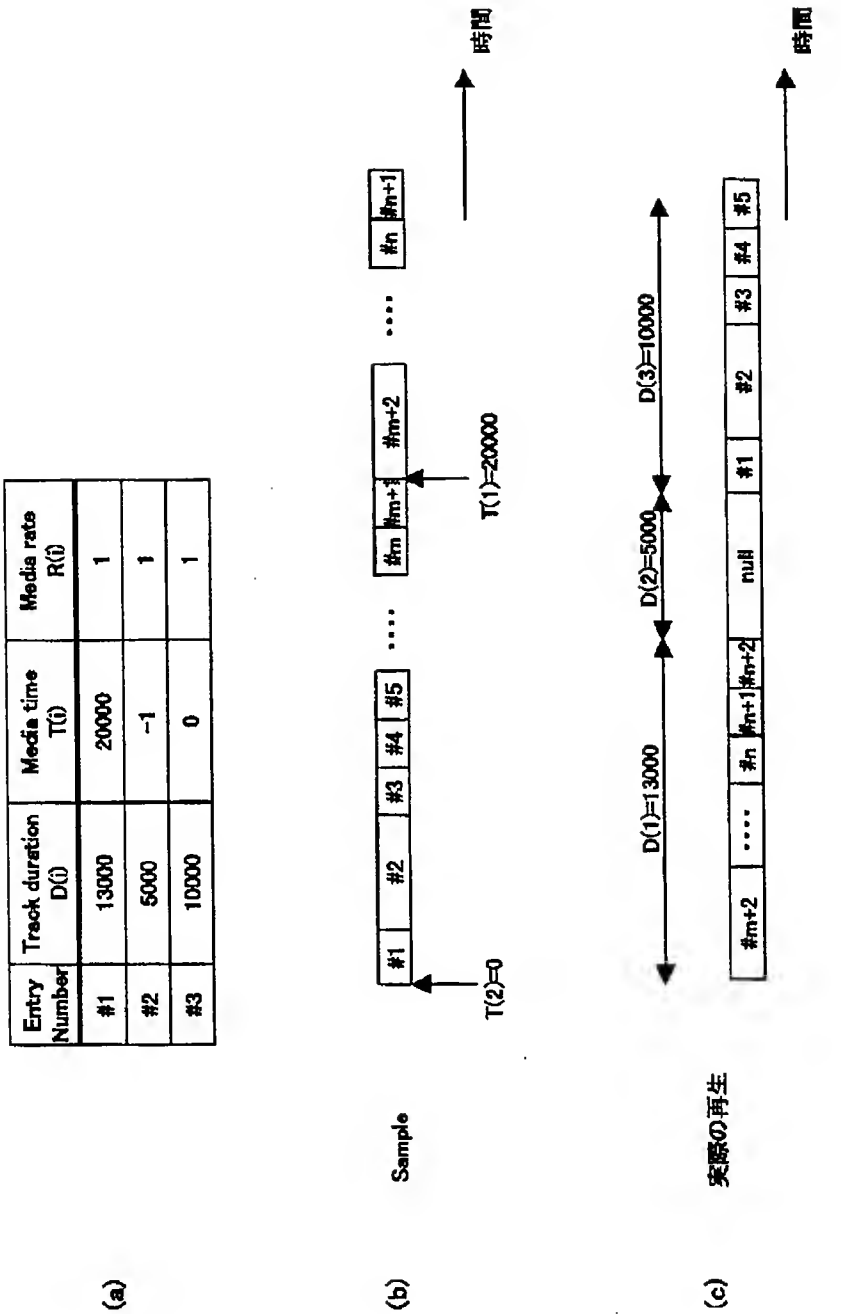
【図９】


ファイル901
chunk#1
chunk#m
chunk#n
chunk offset
sample#j
sample#i
sample size

【図１４】


AVストリーム
Record Unit(RU)#1
Record Unit(RU)#2
Record Unit(RU)#3
Record Unit(RU)#L
Video Unit(VU)#1
Video Unit(VU)#2
Video Unit(VU)#M

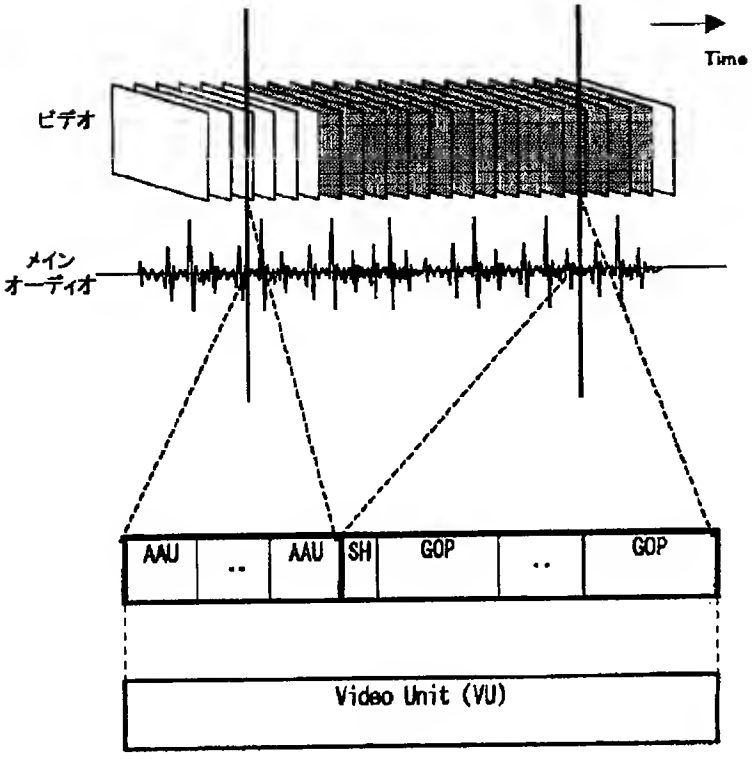
【図１１】



(a)

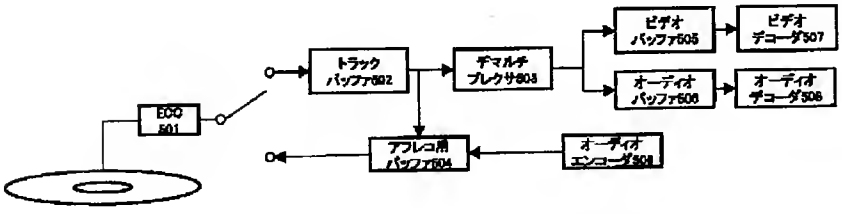
Entry Number	Track duration D(i)	Media time T(i)	Media rate R(i)
#1	13000	20000	1
#2	5000	-1	1
#3	10000	0	1

【図１５】

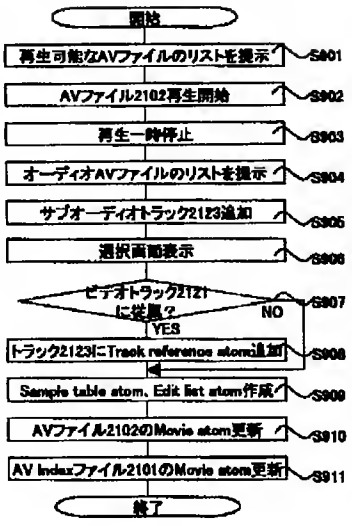


AAU: Audio Access Unit
SH: Sequence Header

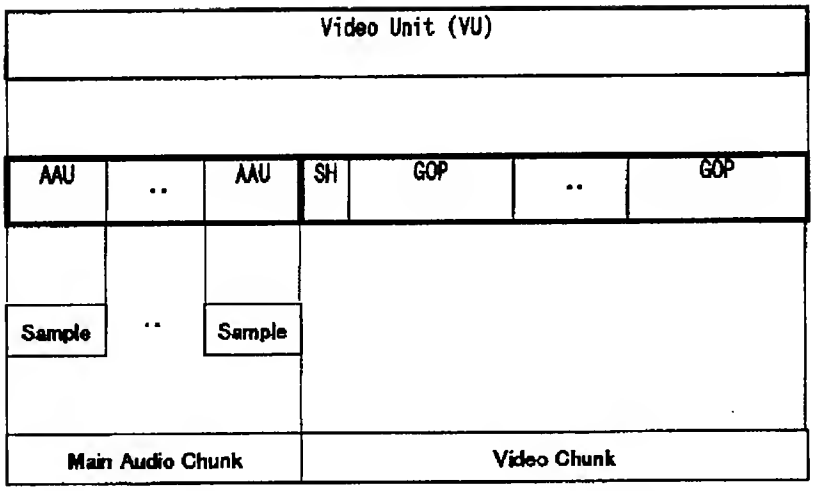
【図１７】



【図２５】



【図１６】

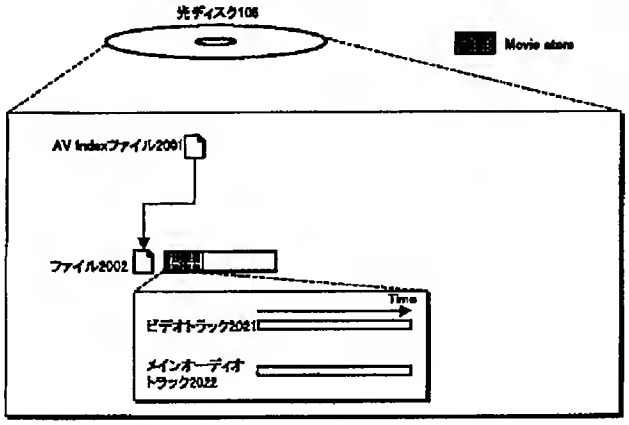


AAU: Audio Access Unit
GOP: Group Of Pictures
SH: Sequence Header

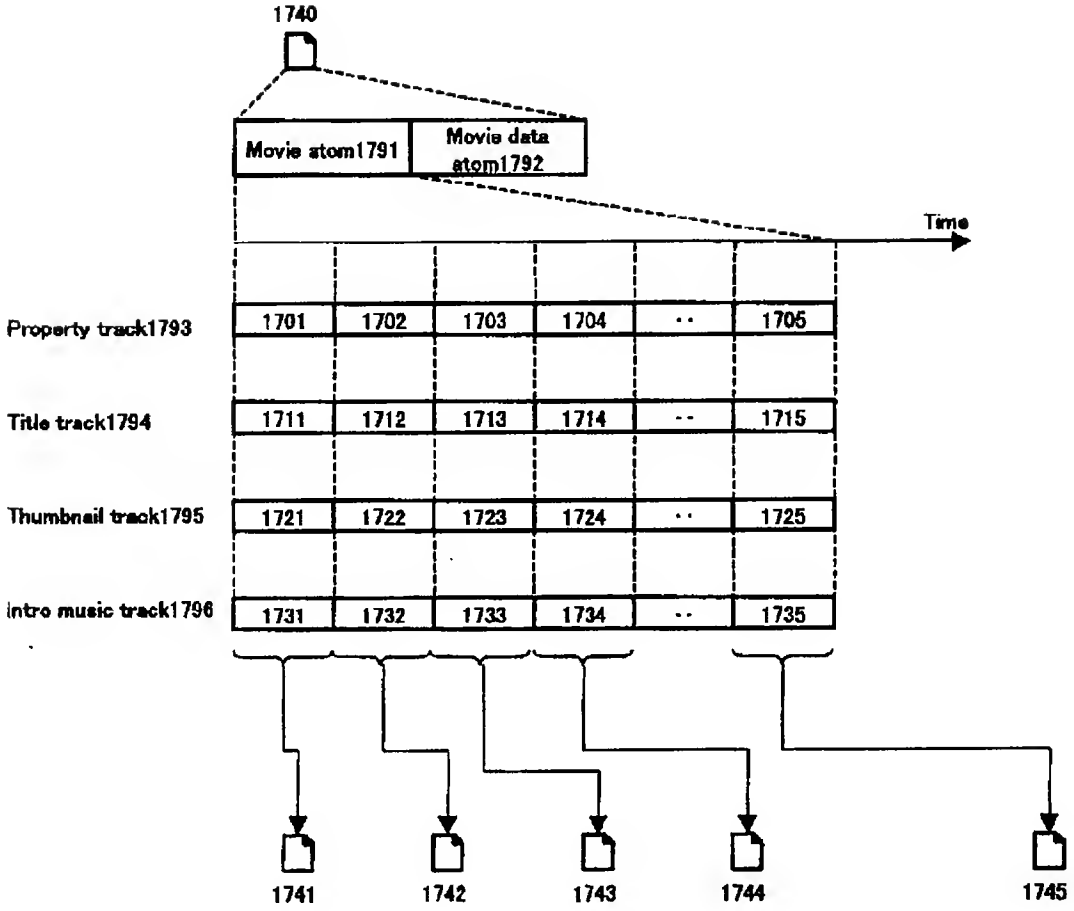
【図１９】

```
Property Entry {
        version
        pe-flags
        parent-entry-number
        entry-number
        set-dependent-flags
        user-private-flags
        reserved
        creation-time
        modification-time
        duration
        binary-file-identifier
        referred-counter
        referring file list
        URL file identifier
}
```

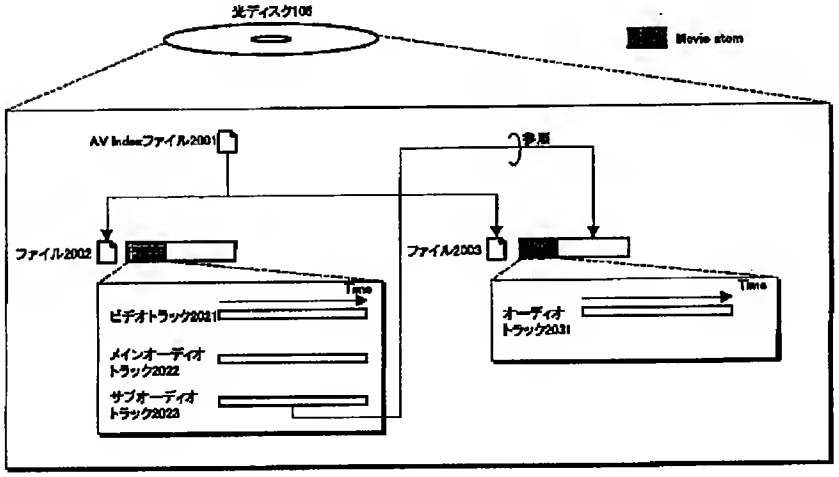
【図２１】



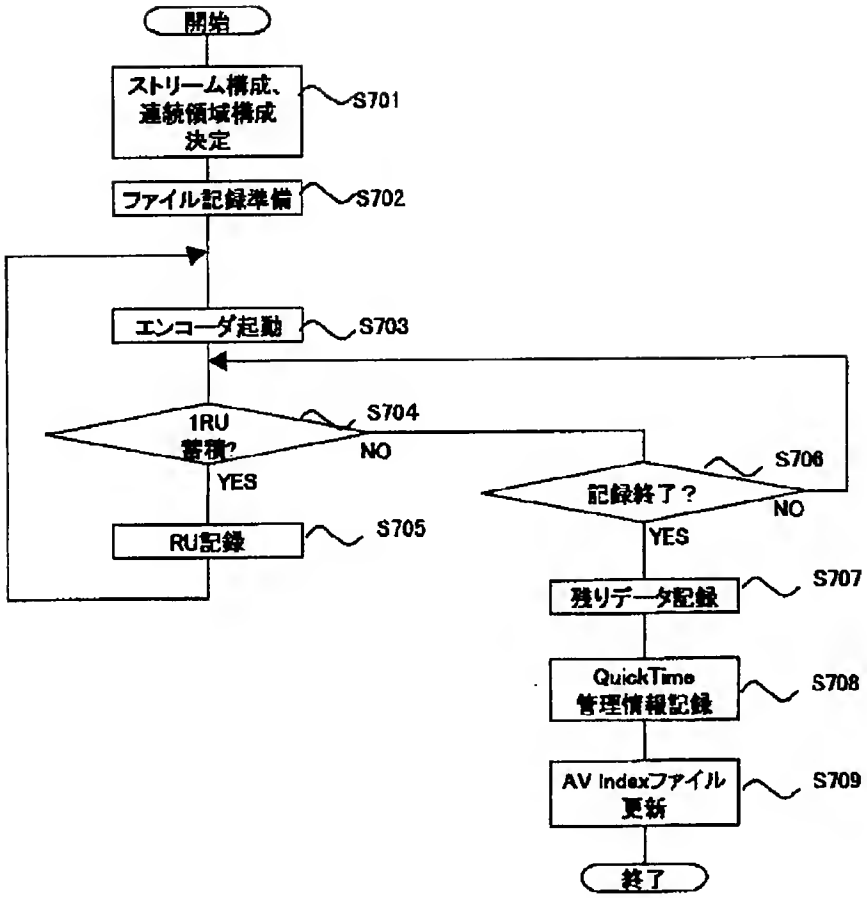
【図１８】


1740
Movie atom1791
Movie data atom1792
Time
Property track1793
Title track1794
Thumbnail track1795
intro music track1796
1701
1702
1703
1704
1705
1711
1712
1713
1714
1715
1721
1722
1723
1724
1725
1731
1732
1733
1734
1735
1741
1742
1743
1744
1745

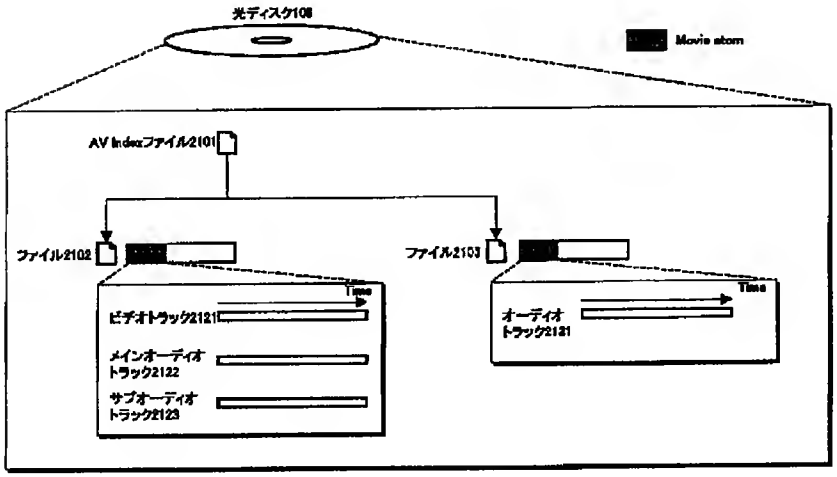
【図２３】


光ディスク106
Movie atom
AV Indexファイル2001
参照
ファイル2002
ファイル2003
Time
ビデオトラック2021
メインオーディオトラック2022
サブオーディオトラック2023
オーディオトラック2031

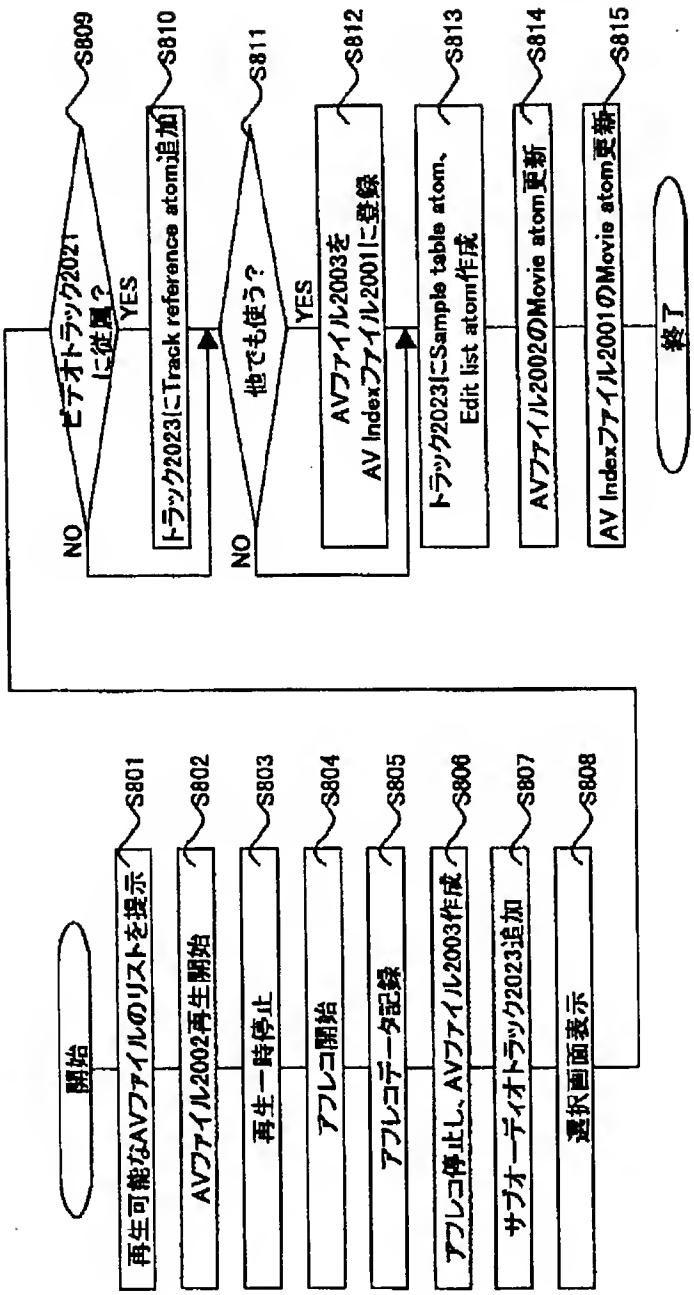
【図２０】


開始
ストリーム構成、連続領域構成決定
S701
ファイル記録準備
S702
エンコーダ起動
S703
1RU蓄積?
S704
NO
YES
RU記録
S705
記録終了？
S706
NO
YES
残りデータ記録
S707
QuickTime管理情報記録
S708
AV Indexファイル更新
S709
終了

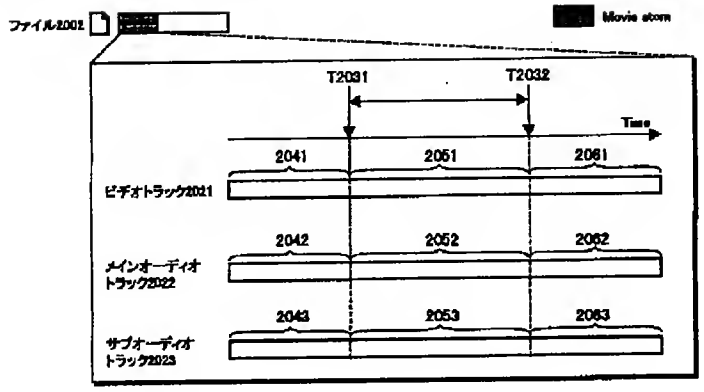
【図２４】


光ディスク108
Movie atom
AV Indexファイル2101
ファイル2102
ファイル2103
Time
ビデオトラック2121
メインオーディオトラック2122
サブオーディオトラック2123
Time
オーディオトラック2131

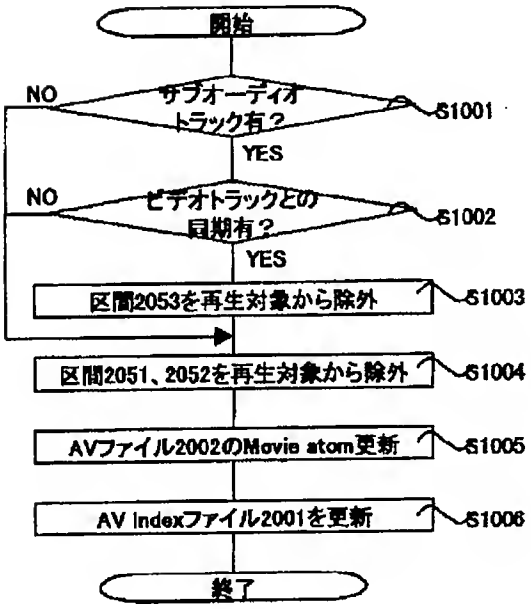
【図２２】


開始
再生可能なAVファイルのリストを提示
S801
AVファイル2002再生開始
S802
再生一時停止
S803
アフレコ開始
S804
アフレコデータ記録
S805
アフレコ停止し、AVファイル2003作成
S806
サブオーディオトラック2023追加
S807
選択画面表示
S808
ビデオトラック2021に従属？
S809
NO
YES
トラック2023にTrack reference atom追加
S810
他でも使う？
S811
NO
YES
AVファイル2003をAV Indexファイル2001に登録
S812
トラック2023にSample table atom、Edit list atom作成
S813
AVファイル2002のMovie atom更新
S814
AV Indexファイル2001のMovie atom更新
S815
終了

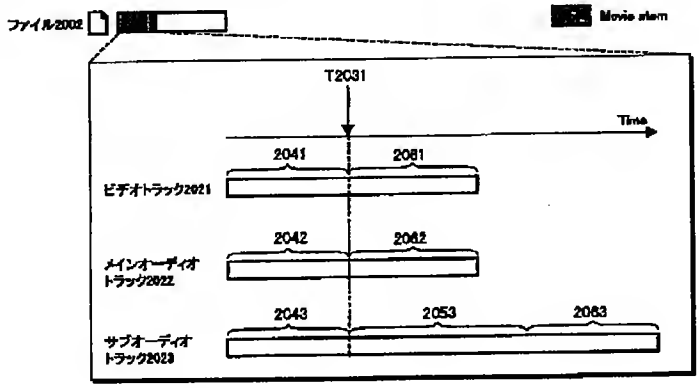
【図２６】


ファイル2002
Movie atom
T2031
T2032
Time
2041
2051
2061
ビデオトラック2021
2042
2052
2062
メインオーディオ
トラック2022
2043
2053
2063
サブオーディオ
トラック2023

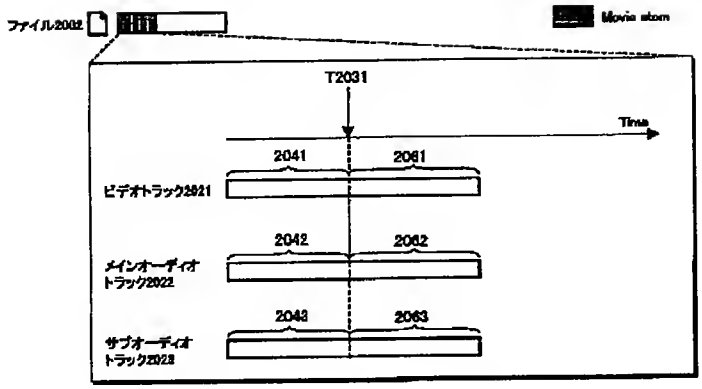
【図２７】


開始
NO
サブオーディオ
トラック有？
S1001
YES
NO
ビデオトラックとの
同期有？
S1002
YES
区間2053を再生対象から除外
S1003
区間2051、2052を再生対象から除外
S1004
AVファイル2002のMovie atom更新
S1005
AV Indexファイル2001を更新
S1006
終了

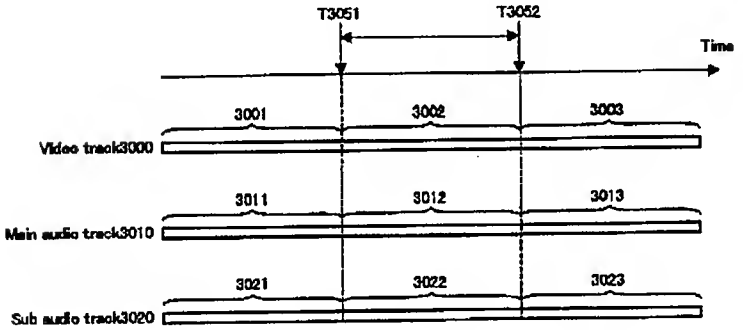
【図２８】


ファイル2002
Movie atom
T2031
Time
2041
2061
ビデオトラック2021
2042
2062
メインオーディオ
トラック2022
2043
2053
2063
サブオーディオ
トラック2023

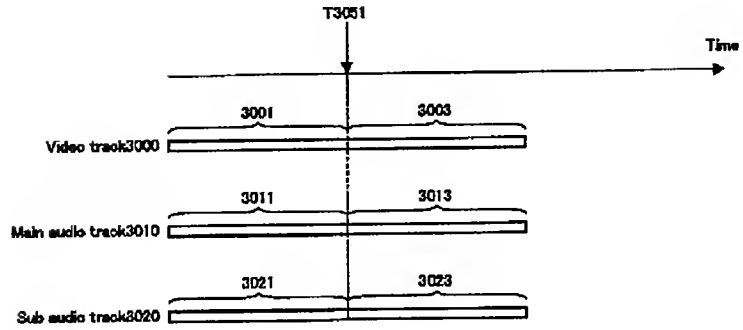
【図２９】


ファイル2002
Movie atom
T2031
Time
2041
2061
ビデオトラック2021
2042
2062
メインオーディオ
トラック2022
2043
2063
サブオーディオ
トラック2023

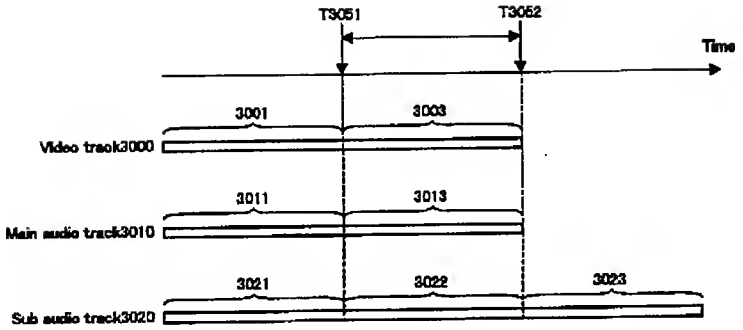
【図３０】


T3051
T3052
Time
3001
3002
3003
Video track3000
3011
3012
3013
Main audio track3010
3021
3022
3023
Sub audio track3020

【図３１】


T3051
Time
3001
3003
Video track3000
3011
3013
Main audio track3010
3021
3023
Sub audio track3020

【図３２】



フロントページの続き

(72)発明者　山口　孝好
大阪府大阪市阿倍野区長池町22番22号　シャープ株式会社内

Ｆターム(参考)　5C053 FA23 GA11 GB15 GB37 HA30 JA07 JA22 JA30
5D044 AB07 BC04 CC06 DE32 DE48 GK12 GM21 HL16
5D110 AA14 AA29 BB01 BB20 CA05 CA06 CA31 CB08 CC06 CF21 DA12 DB03 DC05 DC16 DE01